Amendments to the Claims:

The listing of the claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (Previously Presented): Method for determining a two-dimensional representation of a three-dimensional world, whereby the two-dimensional representation is specified at least by a geometric description, a tree structure for spatial division of the three-dimensional world, and the material properties of all the objects of this world, wherein a ray tracing method is used, whereby several rays are brought together into a packet of discrete rays, whereby an association of a data structure with each packet of discrete rays is carried out, whereby a status-related assignment of the data structure to each ray is stored in memory, whereby all of the following are applied to the packet(s) of the several discrete rays, in that in a work step that performs an operation on a packet, this operation is performed on every ray of the packet, if the status-related assignment is stored in the data structure assigned to the packet, that the ray in question participates in this operation, whereby the ray tracing method is broken down into the following

work steps, which are worked off one after the other, for the individual packets of rays:

A Camera ray generation: A packet of rays is generated, which are supposed to be intersected with the geometry of the three-dimensional world, proceeding from the virtual camera,

B Tree traversal: In accordance with the spatial orientation of the rays, the tree data structure for the geometry of the three-dimensional world is run through, until a tree leaf or a placeholder is reached, whereby in this manner, tree nodes and placeholders are read out of a memory, and calculations are carried out, whereby when a placeholder is reached, the corresponding sequence of instructions of the substitute object is carried out, in that the placeholder is replaced with an object that can also be a complex one, whereby if a placeholder is present, two alternatives are available:

- First, the sequence of instructions is carried out completely, and subsequently, the method is continued in accordance with the new structure of the tree, or
- the sequence of instructions is carried out asynchronously, whereby the method is continued

directly with the substitute of the placeholder, and when the sequence of instructions of the substitute object has been completely worked off, the substitute is replaced with the result of this work,

C Read-in of the tree leaf: All the references of the objects and placeholders that are located in this tree leaf are read in from a memory, and carried out, if applicable,

D Intersecting of all objects: All the objects that were specified in work step C are read in, and intersected with the packet of rays, whereby if a transformation was specified for an object, this is applied before the intersection with the object is calculated, whereby if a geometry-changing sequence of instructions was specified for an object, this is carried out before the intersection with the object is calculated, whereby if an object to be intersected is a complex object having its own tree structure, which must be traversed, this object is intersected in that the work steps B - F, adapted to the complex object, are carried out,

E Intersection evaluation: It is checked whether valid intersection points exist for the packet of rays, hereby possible ray intersections are determined for

elements of the tree for which a ray intersection is possible, but which have not yet been examined, if valid intersection points were not yet determined in a sufficient amount, in that the method starts again at step B, including the previous results, whereby in the case of a sufficient amount of determined ray intersections, the method continues with step F,

F Determining the object properties: With evaluation of the results of the preceding work steps, data of objects with which valid intersection points were calculated, if there are any, are read out of the memory, whereby the results of the calculations flow into step G, whereby if the results are appropriate, values in a memory are recalculated, whereby in the case that a material-changing sequence of instructions was specified for one or more of the objects that were intersected by a ray, in such a manner that the material properties of this object are of significance for the work step F or G, this sequence of instructions is carried out ahead of the calculations of the work step F,

G Subsequent ray generation: If it was calculated in step F that one or more additional rays are supposed to be

intersected with the geometry of the three-dimensional world, then corresponding packets of rays are calculated in this step, and the sequence of the method starts again at work step B for these rays, whereby otherwise, it starts again at step A, if additional camera rays still have to be generated for forming the two-dimensional representation.

Claim 2 (Previously Presented): Method according to claim 1, wherein the rays generated in work step G are brought together in new packets by means of an additional work step, whereby all the newly generated rays are collected, sorted, and compiled into new packets of discrete rays.

Claim 3 (Previously Presented): Method according to claim 1, wherein a multi-threading method is used, in that a packet of rays is considered as a thread, in each instance.

Claim 4 (Previously Presented): Method according to claim 1, wherein the amount of the data of the three-dimensional world is managed in such a manner that only a part of these data is held in the memory at a particular time.

Claim 5 (Currently Amended): ~~Device~~ A device for implementing ~~the~~ a method ~~of claim 1~~ for determining a two-dimensional representation of a three-dimensional world, whereby the two-dimensional representation is specified at least by a geometric description, a tree structure for spatial division of the three-dimensional world, and the material properties of all the objects of this world, wherein a ray tracing method is used, whereby several rays are brought together into a packet of discrete rays, whereby an association of a data structure with each packet of discrete rays is carried out, whereby a status-related assignment of the data structure to each ray is stored in memory, whereby all of the following are applied to the packet(s) of the several discrete rays, in that in a work step that performs an operation on a packet, this operation is performed on every ray of the packet, if the status-related assignment is stored in the data structure assigned to the packet, that the ray in question participates in this operation, whereby the ray tracing method is broken down into the following work steps, which are worked off one after the other, for the individual packets of rays:

A Camera ray generation: A packet of rays is generated, which are supposed to be intersected with the geometry

of the three-dimensional world, proceeding from the virtual camera,

B Tree traversal: In accordance with the spatial orientation of the rays, the tree data structure for the geometry of the three-dimensional world is run through, until a tree leaf or a placeholder is reached, whereby in this manner, tree nodes and placeholders are read out of a memory, and calculations are carried out, whereby when a placeholder is reached, the corresponding sequence of instructions of the substitute object is carried out, in that the placeholder is replaced with an object that can also be a complex one, whereby if a placeholder is present, two alternatives are available:

- First, the sequence of instructions is carried out completely, and subsequently, the method is continued in accordance with the new structure of the tree, or
- the sequence of instructions is carried out asynchronously, whereby the method is continued directly with the substitute of the placeholder, and when the sequence of instructions of the substitute object has been completely worked off, the substitute is replaced with the result of this work,

C Read-in of the tree leaf: All the references of the objects and placeholders that are located in this tree leaf are read in from a memory, and carried out, if applicable,

D Intersecting of all objects: All the objects that were specified in work step C are read in, and intersected with the packet of rays, whereby if a transformation was specified for an object, this is applied before the intersection with the object is calculated, whereby if a geometry-changing sequence of instructions was specified for an object, this is carried out before the intersection with the object is calculated, whereby if an object to be intersected is a complex object having its own tree structure, which must be traversed, this object is intersected in that the work steps B - F, adapted to the complex object, are carried out,

E Intersection evaluation: It is checked whether valid intersection points exist for the packet of rays, hereby possible ray intersections are determined for elements of the tree for which a ray intersection is possible, but which have not yet been examined, if valid intersection points were not yet determined in a sufficient amount, in that the method starts again at

step B, including the previous results, whereby in the case of a sufficient amount of determined ray intersections, the method continues with step F,

F Determining the object properties: With evaluation of the results of the preceding work steps, data of objects with which valid intersection points were calculated, if there are any, are read out of the memory, whereby the results of the calculations flow into step G, whereby if the results are appropriate, values in a memory are recalculated, whereby in the case that a material-changing sequence of instructions was specified for one or more of the objects that were intersected by a ray, in such a manner that the material properties of this object are of significance for the work step F or G, this sequence of instructions is carried out ahead of the calculations of the work step F,

G Subsequent ray generation: If it was calculated in step F that one or more additional rays are supposed to be intersected with the geometry of the three-dimensional world, then corresponding packets of rays are calculated in this step, and the sequence of the method starts again at work step B for these rays, whereby

otherwise, it starts again at step A, if additional camera rays still have to be generated for forming the two-dimensional representation, wherein the individual work steps are represented by independent functional groups.

Claim 6 (Currently Amended): ~~Device~~ A device for implementing ~~the~~ a method ~~of claim 1~~ for determining a two-dimensional representation of a three-dimensional world, whereby the two-dimensional representation is specified at least by a geometric description, a tree structure for spatial division of the three-dimensional world, and the material properties of all the objects of this world, wherein a ray tracing method is used, whereby several rays are brought together into a packet of discrete rays, whereby an association of a data structure with each packet of discrete rays is carried out, whereby a status-related assignment of the data structure to each ray is stored in memory, whereby all of the following are applied to the packet(s) of the several discrete rays, in that in a work step that performs an operation on a packet, this operation is performed on every ray of the packet, if the status-related assignment is stored in the data structure assigned to the packet, that the ray in question participates in this operation,

whereby the ray tracing method is broken down into the following work steps, which are worked off one after the other, for the individual packets of rays:

A Camera ray generation: A packet of rays is generated, which are supposed to be intersected with the geometry of the three-dimensional world, proceeding from the virtual camera,

B Tree traversal: In accordance with the spatial orientation of the rays, the tree data structure for the geometry of the three-dimensional world is run through, until a tree leaf or a placeholder is reached, whereby in this manner, tree nodes and placeholders are read out of a memory, and calculations are carried out, whereby when a placeholder is reached, the corresponding sequence of instructions of the substitute object is carried out, in that the placeholder is replaced with an object that can also be a complex one, whereby if a placeholder is present, two alternatives are available:

- First, the sequence of instructions is carried out completely, and subsequently, the method is continued in accordance with the new structure of the tree, or

• the sequence of instructions is carried out asynchronously, whereby the method is continued directly with the substitute of the placeholder, and when the sequence of instructions of the substitute object has been completely worked off, the substitute is replaced with the result of this work,

C Read-in of the tree leaf: All the references of the objects and placeholders that are located in this tree leaf are read in from a memory, and carried out, if applicable,

D Intersecting of all objects: All the objects that were specified in work step C are read in, and intersected with the packet of rays, whereby if a transformation was specified for an object, this is applied before the intersection with the object is calculated, whereby if a geometry-changing sequence of instructions was specified for an object, this is carried out before the intersection with the object is calculated, whereby if an object to be intersected is a complex object having its own tree structure, which must be traversed, this object is intersected in that the work steps B - F, adapted to the complex object, are carried out,

E Intersection evaluation: It is checked whether valid intersection points exist for the packet of rays, hereby possible ray intersections are determined for elements of the tree for which a ray intersection is possible, but which have not yet been examined, if valid intersection points were not yet determined in a sufficient amount, in that the method starts again at step B, including the previous results, whereby in the case of a sufficient amount of determined ray intersections, the method continues with step F,

F Determining the object properties: With evaluation of the results of the preceding work steps, data of objects with which valid intersection points were calculated, if there are any, are read out of the memory, whereby the results of the calculations flow into step G, whereby if the results are appropriate, values in a memory are recalculated, whereby in the case that a material-changing sequence of instructions was specified for one or more of the objects that were intersected by a ray, in such a manner that the material properties of this object are of significance for the work step F or G, this sequence of instructions

is carried out ahead of the calculations of the work step F,

G Subsequent ray generation: If it was calculated in step F that one or more additional rays are supposed to be intersected with the geometry of the three-dimensional world, then corresponding packets of rays are calculated in this step, and the sequence of the method starts again at work step B for these rays, whereby otherwise, it starts again at step A, if additional camera rays still have to be generated for forming the two-dimensional representation,

wherein the work steps are brought together in functional groups, as follows:

- RGS: Comprises several functional units for the work steps A, F, and G, and expands them with units for the management of sub-units.

  - Master: Coordinates the work of the slave functional units.

  - Slave: Contains units for the work steps A, F, and G.

  - MemInt: Coordinates the access of slave units to the external memory.

- RTC: Comprises several functional units for the work steps B, C, D, and E.
  - Traversal: Functional unit that comprises the work steps B and E.
  - List: Functional unit for the work step C.
  - Intersection: Functional unit for the work step D.
- RTC-MI: Coordinates and regulates all accesses of the RTC units to external memory units.
  - T-SR: Coordinates the access of traversal units to the T-cache.
  - T-cache: Keeps book about preceding traversal accesses and tries to avoid new accesses, in that it re-uses the preceding results.
  - L-SR: Coordinates the access of list units to the L-cache.
  - L-cache: Analogous to the T-cache for list accesses.
  - I-SR: Coordinates the access of list units to the I-cache.
  - I-cache: Analogous to the T-cache for intersection accesses.
  - M-SR: Coordinates the access of-cache units to the external memory.

- MemCtrl: Passes the accesses on to the external memory responsible for them.